AlgoExpert Quad Layout Java 14px Emacs Monokai

Prompt

Sample Output (for creation)

```
The structure below is the root of the trie.
{
  "c": {"*": true},
  "b": {
    "c": {"*": true},
    "a": {"b": {"c": {"*": true}}},
  },
  "a": {"b": {"c": {"*": true}}},
}
```

Sample Input (for searching in the suffix trie above)

```
string = "abc"
```

Sample Output (for searching in the suffix trie above)

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```
      populateSuffixTrieFrom(i, str);
    }
  }

  private void populateSuffixTrieFrom(int index, String s
    TrieNode currentPointer = root;
    for (int j = index; j < str.length(); j++) {
      char ch = str.charAt(j);
      if (!currentPointer.children.containsKey(ch)) {
        TrieNode newTrieNode=new TrieNode();
        currentPointer.children.put(ch, newTrieNode);
      }
      currentPointer=currentPointer.children.get(ch);
    }
    currentPointer.children.put(endSymbol, null);
  }
```

Tests

Custom Output Submit Code

✓

Congratulations!

Your submission passed all of our test cases!